

Novel Approximation Algorithm for Rapid Computation of Steiner Tree Problems with Minimal Error Probability

26 December 2022

Simon Edwards

Research Acceleration Initiative

Introduction

Upon request, I look a look at some of the current persistent problems in computer science and while this topic is of minimal interest to me, I know how valuable a fresh perspective from non-indoctrinated outsiders can be for any field. I have come up with the following after reading a bit about the problem of whether there is any way to reduce the computational time for solving a Steiner Tree Problem without introducing any risk of error. I understand that many have already developed so-called approximation algorithms, which was going to be my suggestion if someone had not already done this. It occurred to me that Steiner Tree Optimization has application for A.I. in addition to some of their more mundane uses such as planning efficient routes. The following is a Steiner Tree Approximation approach that as closely as possible mimics human neural function.

Abstract

The long method for computing the most efficient route from A to B, where each line segment is assigned a "cost" value, is to test every possible combination of line segments to find the lowest value. In complex systems consisting of thousands or perhaps millions of line segments, one can only hope to arrive at pathways that are approximate to the most efficient path. There may even be a benefit to occasionally taking a less efficient path. In cognition, unconventional neurologies often give rise to unique and valuable ideas. This has implications for A.I. For route planning, it is to everyone's mutual benefit when all drivers do not take the allegedly most efficient path. Google Maps, for instance, often has to use coin flips to decide whether to send a specific driver down one of two equally viable routes.

If every driver were sent down the same path, the effect would be the reduction of the efficiency of that path. There would be a traffic jam on whichever path Google deemed "most efficient" and no traffic at all on the alternate pathway. Knowing the best route to take may actually create an inefficiency without knowledge of the intentions of other users of a roadway of any sort. This has implications for cognition in addition to routing automobile traffic.

For example, given that we know that given parts of the brain perform multiple functions, one might think it potentially useful to avoid using the brain for other purposes as it might distract that part of the mind. Counterintuitively, recent studies have found that where there is overlap in brain regions in terms of function, engaging in other functions that activate that same brain region actually stimulate cognition rather than posing a distraction. This may be due to the fact that new formulating new ideas requires a shifting in the physical parts of the brain utilized and not a endless

repetition of the same processing steps. Alternating between activities that utilize the same brain regions forces alternative axon activations.

When most people eat an entrée with two sides, they take a few bites of each food item and rotate between the food items rather than eating say, all of the chicken, then all of the rice, then all of the green beans. Varying sensory inputs allows for the greatest enjoyment to be reaped from any experience as changing inputs has the effect of re-sensitizing a person to each input type. For the same reason, focus on a single cognitive task for extended periods leads to inefficient function while adding variety in the form of pacing or gesticulating while cogitating acts as a cognitive multiplier. A person employing such a cognitive strategy would be, to a certain extent, seeing problems with new eyes from one moment to the next.

That said, establishing efficient artificial neural networks that still allow for variation in approach is essential if human cognition is to be duplicated within a machine.

The first step in my strategy for generating approximated solutions to Steiner Tree Problems is actually to first approximate the Least Efficient Pathway. Both when approximating the least and most efficient pathways, weighted dice rolls are used to decide on an arbitrary pathway. If there is a triple fork in the road, for instance, and option 1 has a cost of "1," option 2 has a cost of "2," and option 3 has a cost of "1," then there is a 1 in 6 chance the weighted dice roll will attempt to take the high-cost path, a 2.5 out of 6 chance that it will take option 1, and a 2.5 out of 6 chance that option 3 will be selected.

Both when calculating the least and most efficient pathways, in my system, simulation attempts that run from A to B are run in parallel with attempts to sim B to A. The total number of line segments in the overall system divided by two and with a value of one added to that determines the maximum number of jumps that the sim will take before giving up and trying a different (weighted) randomly selected route. *When a round is completed in which the A-B route meets up with the B-A route, a candidate ideal path is established.*

I mentioned that a candidate least efficient route must be computed before computing the most efficient route. This is because computing a candidate least efficient route allows us to eliminate large numbers of candidate most efficient routes from consideration, accelerating that computational step when the time comes.

For our purposes, we'll restrict from consideration pathways that run through any nodes within two degrees of separation of the mid-course juncture of simulated A-B and B-A least efficient routes. We are only looking at the mid-course position of least efficient pathways since least and most efficient pathways will tend to overlap near points A and B.

By having a system that favors low-cost choices in the near-term but leaves open the possibility of making higher-cost choices and yet restricts from consideration routes that exceed "Total segment count/2+1" while also restricting from consideration routes that are, at mid-course, physically proximal to highly inefficient routes, a series of simulated paths that would

number in the dozens rather than in the millions would enable the generation of candidate most efficient paths with an extremely low probability of error. This system also has the benefit of automatically balancing the factors of route length versus cost.

This system also has as a virtue the ability to run sims repeatedly and compare multiple candidate pathways for improved accuracy. For artificial neural networks, computations performed by two similar but distinct pathways at the same time but with flow of direction being inverted (A-B versus B-A) provides greater cognitive diversity than varying direction or pathway alone. Furthermore, having two pathways that are less efficient than ideal but equally efficient to one another guarantees the continued use of both pathways. If a human brain has established two pathways and one is less efficient than the other, the less efficient pathway will fall into disuse and overall cognitive capacity will decline. Having two nearly-ideal axons performing related work has multiple benefits versus one ideal and one near-ideal or the ideal axon alone, as you will discover.

For example, if a road can wind either to the left or to the right and I am trying to solve for the correct combination of directional changes to solve a combination that is defined by directional changes, then this has certain advantages. In systems where there are generally roughly equal numbers of left and right turns (there would have to be in order for an axon to get to where it's going) the correct combination is more likely to consist of LEFT RIGHT LEFT RIGHT than it is to consist of LEFT LEFT LEFT RIGHT or RIGHT RIGHT RIGHT LEFT.

If my "most efficient pathway" candidate Y is "LEFT RIGHT LEFT RIGHT" structurally and my candidate Z is "LEFT RIGHT RIGHT LEFT" and I take into consideration that signals can flow in both directions, A-B as well as B-A, four distinct combinations can be tried in each iteration rather than the two that could be tried when only a single "absolute most efficient" pathway is used. Neuromorphic processors as well as artificial neural networks both benefit from cognitive diversity in a manner similar to human neural pathways. It should be noted that cognition is most efficient when alternate pathways, even when they lead to the same destination, while both independently efficient, do not share in common convolutional patterns in common at any point in the pathway. This may explain why cognition in the human brain is supported by astrocytes which lead to a common point (the axon) and why each arm of an astrocyte is roughly identical in length but randomized in terms of the patterns of left, right, up, and down twists. A wider variety of convolutional twists in astrocytes as well as greater numbers of arms on astrocytes generally produces superior cognitive capacity. In some cases, it should be noted, the presence of an excessive number of arms on a single astrocyte can diminish cognitive capacity. This may be a result of crowding of the arms diminishing the variety of convolutional shapes that the fibrous arms may be formed into i.e. this causes them to look more like sine waves than stock market line graphs.

Even in the case of choosing the ideal route for running fiber-optic cable or building highways, given that the absolute most efficient route may change with time, choosing an off-center route may ultimately be more efficient since

it leaves open the option to build an additional road on the other side of the true "most efficient" path so that the average of the two paths works out to be effectively equivalent to the ideal. This may even be preferable where there is a desire to create redundancy and not to put too many eggs in a single proverbial basket. When planning for future expansion of road systems of any sort, when the nature of the future change is unknown, the most efficient choice is not to overcommit to any one direction. Much more efficient is to build multiple two-lane highways a few miles apart than to try to guess at a future need incorrectly, building a four-lane highway in a less-than ideal location. This principle forms a fourth-dimensional corollary to Hypotenuse Theorem.

Conclusion

While many computer science problems may seem as trivial as ever, I found the path optimization challenge to have wider-ranging implications than I had initially anticipated and have therefore given it due consideration, culminating in the aforementioned analysis.